

Nice Neighbour Pharmacy

FIT5032 Assignment 2

Student Name: **Hao Xu**

Student ID: **32767919**

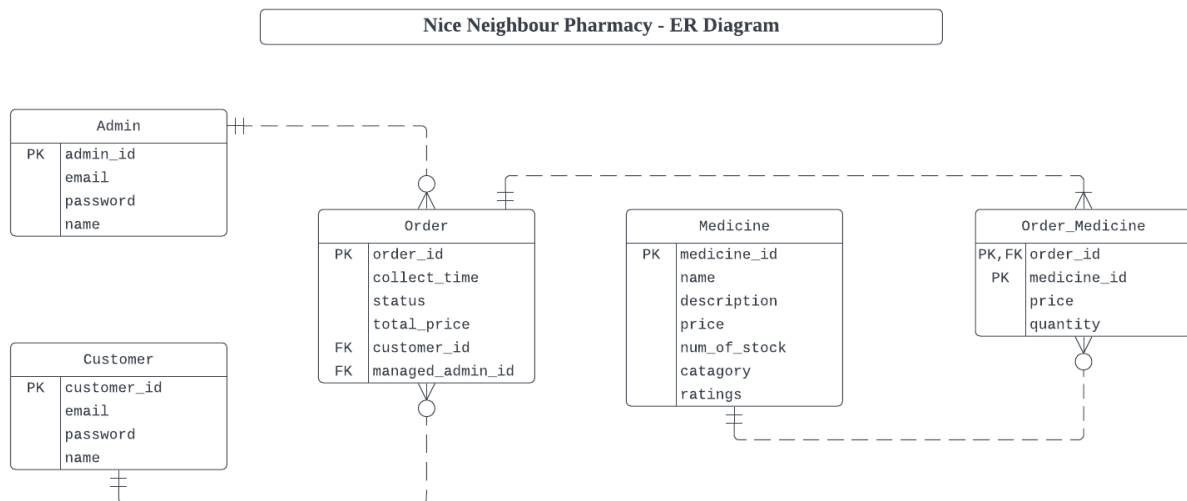
GitHub Link for the Project:

<https://github.com/BrantleyXU/FIT5032/tree/main/portfolio/NiceNeighbourPharmacy>

1. Entity Relationship Diagram

In this ER Diagram, at least 4 business requirements are related:

1. B.1.: Authentication: only authenticated customers are allowed to order medicines. Otherwise, they cannot see “Medicine” pages.
2. C.1.: Role based authentication: there are 2 roles in the application, admins and customers. Customers can order medicines; however, admins can see all orders from all customers.
3. D.3.: Rating: in the “medicine” entity, there is a field called “ratings” which is a rate score given by all customers who have already bought and used this medicine.
4. D.4.: Booking constraint: in the “order” entity, there is a field called “collect_date” which is a collecting date chosen by customers. The constraint is that customers cannot choose the same date and time as previous customers.

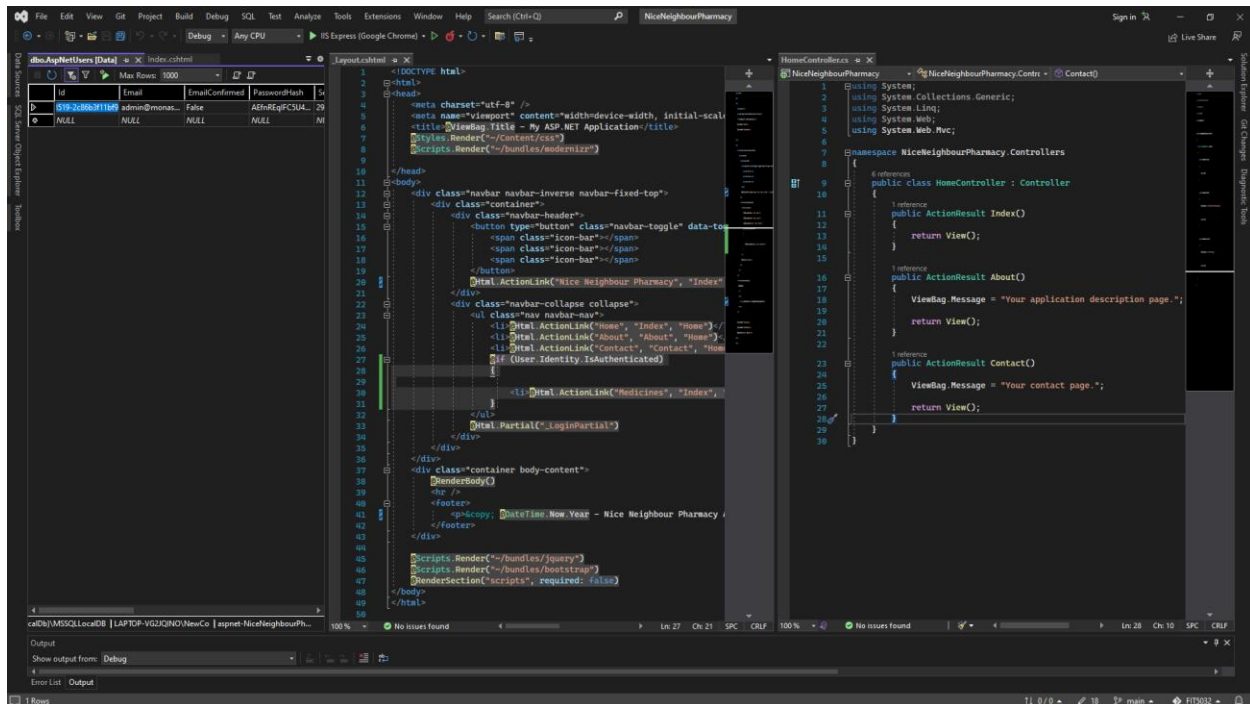


2. Home Controller and User Authentication

GitHub Link for HomeController:

<https://github.com/BrantleyXU/FIT5032/blob/main/portfolio/NiceNeighbourPharmacy/NiceNeighbourPharmacy/Controllers/HomeController.cs>

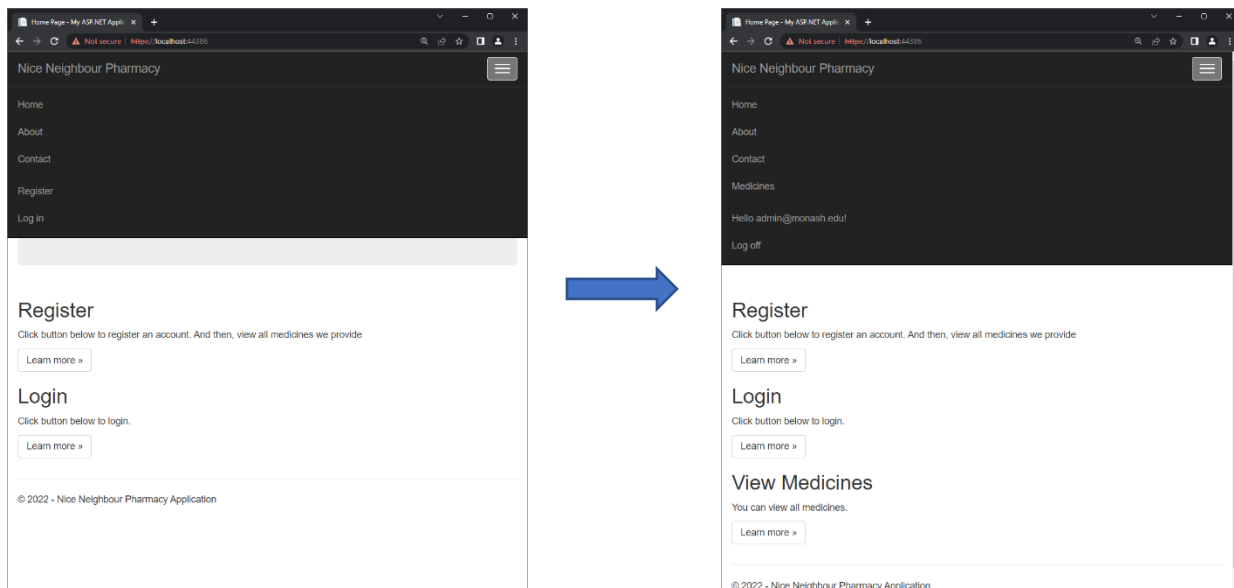
The following screenshot shows the HomeController and User Authentication methods. In this application, a new customer can register an account. After that, customers can login. Only authenticated customers can see “Medicines” webpage.



The screenshot shows the Visual Studio IDE with the following components:

- Database Explorer:** Shows a table named 'Users' with columns: Id, Email, EmailConfirmed, PasswordHash, and IsLockedOut. The table has 1000 rows.
- Layout.cshtml:** Shows the HTML structure of the application, including the navigation bar with links for Home, About, Contact, Register, and Login. The 'Medicines' link is only visible if the user is authenticated.
- HomeController.cs:** Shows the implementation of the HomeController, including the Index, About, and Contact methods. The Index method returns the 'Index' view.
- AccountController.cs:** Shows the implementation of the AccountController, including the Register, Login, and Logout methods. The Register method returns the 'Register' view.

The following screenshot shows that, at the front-end, a new customer cannot see “Medicine” page. After he/she login, “Medicine” page option will be displayed.

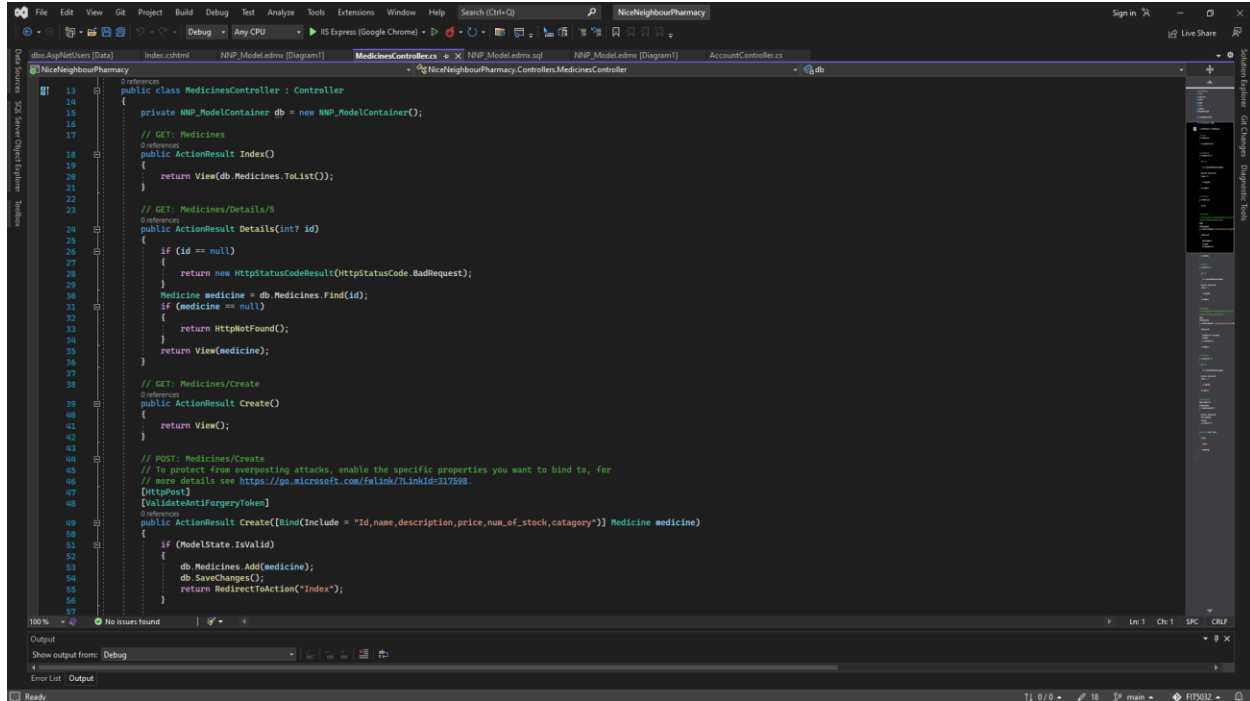


3. Other Controller and Usability Evaluation

GitHub Link for MedicinesController:

<https://github.com/BrantleyXU/FIT5032/tree/main/portfolio/NiceNeighbourPharmacy/NiceNeighbourPharmacy/Controllers>

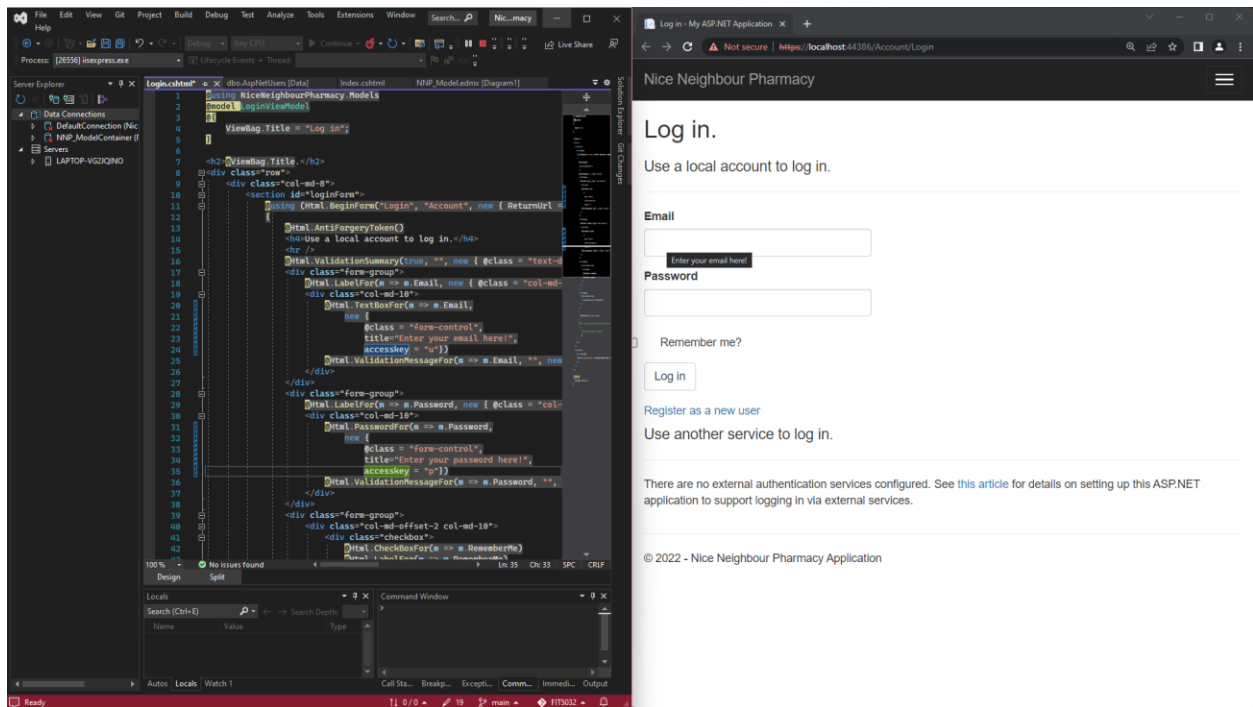
The following screenshot shows another Controller in this application, which is “MedicineController”. This controller is used for adding new medicine, updating medicines’ details and deleting medicines by admins, and searching medicines based on some conditions by customers.



```
13 public class MedicinesController : Controller
14 {
15     private NNP_ModelContainer db = new NNP_ModelContainer();
16
17     // GET: Medicines
18     public ActionResult Index()
19     {
20         return View(db.Medicines.ToList());
21     }
22
23     // GET: Medicines/Details/5
24     public ActionResult Details(int? id)
25     {
26         if (id == null)
27         {
28             return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
29         }
30         Medicine medicine = db.Medicines.Find(id);
31         if (medicine == null)
32         {
33             return HttpNotFound();
34         }
35         return View(medicine);
36     }
37
38     // GET: Medicines/Create
39     public ActionResult Create()
40     {
41         return View();
42     }
43
44     // POST: Medicines/Create
45     // To protect from overposting attacks, enable the specific properties you want to bind to, for
46     // more details see https://go.microsoft.com/fwlink/?linkid=317598
47     [HttpPost]
48     [ValidateAntiForgeryToken]
49     public ActionResult Create([Bind(Include = "Id,name,description,price,num_of_stock,category")] Medicine medicine)
50     {
51         if (ModelState.IsValid)
52         {
53             db.Medicines.Add(medicine);
54             db.SaveChanges();
55             return RedirectToAction("Index");
56         }
57     }
58 }
```

Usability Evaluation

According to Offutt (2002), usability is considered as one of the most important quality factors for web applications. Thus, usability evaluation becomes more critical in ensuring the success of web applications (Neuwirth & Regli, 2002). In this application, multiple approaches are applied to enhance usability design. The first one is called “Tool Tips” (User3559349, 2015). In the right side of the screenshot, when users hover their mouse above the text input field, a tip saying, “Enter your email here!”, will be displayed, so that users will know what should be entered. So as “Password” text input field. The second one is called “Access Hot Keys”. In the left side of the screenshot, hot keys “u” and “p” are assigned to “Email” and “Password”, respectively. So, when users press “Alt+u”, the cursor will automatically go to “Email”, so that users do not need to use mouse to click elements on the website. The final enhancement is that users could use “Tab” button to navigate among different sections of the webpage, which is set as default.



References

- Offutt, J. (2002). Quality attributes of web software applications. *IEEE software*, 19(2), 25-32.
- Neuwirth, C. M., & Regli, S. H. (2002). Guest Editors' Introduction: Usability and the Web. *IEEE Internet Computing*, 6(2), 44.
- User3559349. (2015, June 9). Adding html attributes using `Html.EditorFor()`. [Answer to the online question *Display a tooltip with Html helper.*]. Stack Overflow.
<https://stackoverflow.com/a/30743880>